

In [1]:

```
#importing all the libraries.
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [3]:

```
#importing and readding the dataset
dataset = pd.read_csv("dataset/Superstore.csv")
```

In [4]:

```
#the first five values in the dataset
dataset.head()
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20

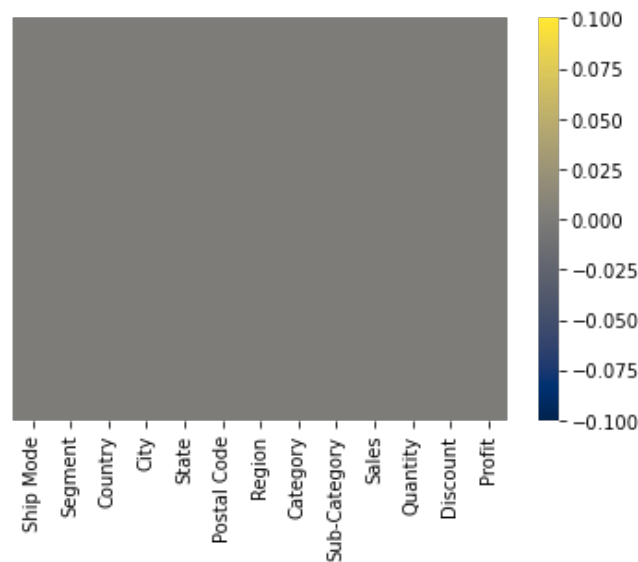
## Data Cleaning

In [9]:

```
#Dealing with the Missing Data in the dataset
sns.heatmap(dataset.isnull(), yticklabels=False, cbar=True, cmap='cividis')
```

Out[9]:

<AxesSubplot:>



In [10]:

```
#number of rows and columns
dataset.shape
```

Out[10]:

(9994, 13)

In [11]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Ship Mode       9994 non-null   object
 1   Segment         9994 non-null   object
 2   Country         9994 non-null   object
 3   City            9994 non-null   object
 4   State          9994 non-null   object
 5   Postal Code     9994 non-null   int64
 6   Region         9994 non-null   object
 7   Category       9994 non-null   object
 8   Sub-Category   9994 non-null   object
 9   Sales          9994 non-null   float64
10  Quantity       9994 non-null   int64
11  Discount       9994 non-null   float64
12  Profit         9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
```

In [12]:

```
dataset.describe()
```

Out[12]:

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

In [13]:

```
#checking in the Category feild.
dataset.groupby(["Category"]).mean()
```

Out[13]:

	Postal Code	Sales	Quantity	Discount	Profit
Category					
Furniture	55726.556341	349.834887	3.785007	0.173923	8.699327
Office Supplies	54890.951211	119.324101	3.801195	0.157285	20.327050
Technology	55551.572279	452.709276	3.756903	0.132323	78.752002

In [14]:

```
#checking in the Category field.
dataset.groupby(["Category"]).max()
```

Out[14]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Sub-Category	Sales	Quantity	Discount	Profit
Category												
Furniture	Standard Class	Home Office	United States	York	Wyoming	99301	West	Tables	4416.174	14	0.7	1013.11
Office Supplies	Standard Class	Home Office	United States	Yuma	Wisconsin	99301	West	Supplies	9892.740	14	0.8	4946.31
Technology	Standard Class	Home Office	United States	Yuma	Wisconsin	99207	West	Phones	22638.480	14	0.7	8399.97

In [15]:

```
#checking in the Category field.
dataset.groupby(["Category"]).min()
```

Out[15]:

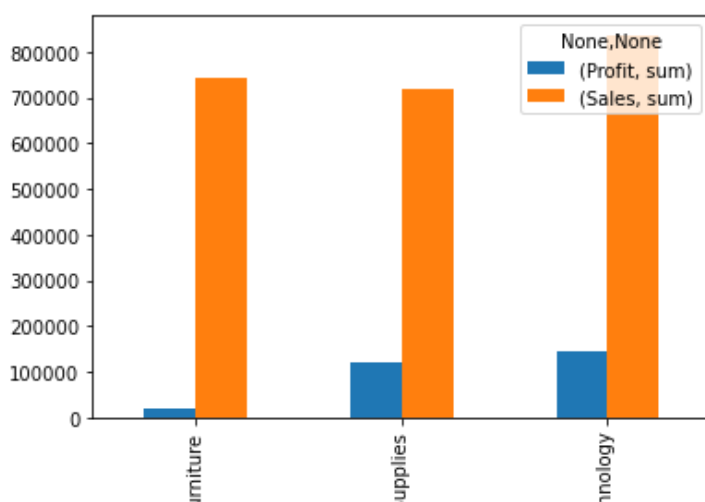
	Ship Mode	Segment	Country	City	State	Postal Code	Region	Sub-Category	Sales	Quantity	Discount	Profit
Category												
Furniture	First Class	Consumer	United States	Akron	Alabama	1040	Central	Bookcases	1.892	1	0.0	1862.31
Office Supplies	First Class	Consumer	United States	Aberdeen	Alabama	1453	Central	Appliances	0.444	1	0.0	3701.89
Technology	First Class	Consumer	United States	Akron	Alabama	1841	Central	Accessories	0.990	1	0.0	6599.97

In [16]:

```
#Finding the totalsum and visualising the comparison of total profit with respect to the sales.
plt.figure(figsize= (10,20))
dataset.groupby('Category')['Profit','Sales'].agg(['sum']).plot.bar()
plt.show()
```

```
<ipython-input-16-ca19f90f2262>:3: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
dataset.groupby('Category')['Profit','Sales'].agg(['sum']).plot.bar()
```

<Figure size 720x1440 with 0 Axes>



In [17]:

```
#Checking in the sub category field
dataset.groupby('Sub-Category').mean()
```

Out[17]:

	Postal Code	Sales	Quantity	Discount	Profit
Sub-Category					
Accessories	57378.624516	215.974604	3.840000	0.078452	54.111788
Appliances	54185.703863	230.755710	3.710300	0.166524	38.922758
Art	54434.243719	34.068834	3.768844	0.074874	8.200737
Binders	54908.994091	133.560560	3.922521	0.372292	19.843574
Bookcases	56015.521930	503.859633	3.807018	0.211140	-15.230509
Chairs	56622.737439	532.332420	3.818476	0.170178	43.095894
Copiers	56962.897059	2198.941618	3.441176	0.161765	817.909190
Envelopes	52463.507874	64.867724	3.566929	0.080315	27.418019
Fasteners	57631.626728	13.936774	4.211982	0.082028	4.375660
Furnishings	54211.525601	95.825668	3.723093	0.138349	13.645918
Labels	53716.991758	34.303055	3.846154	0.068681	15.236962
Machines	55344.939130	1645.553313	3.826087	0.306087	29.432669
Paper	55692.862044	57.284092	3.779562	0.074891	24.856620
Phones	53877.587177	371.211534	3.699663	0.154556	50.073938
Storage	54667.517730	264.590553	3.732861	0.074704	25.152277
Supplies	55966.094737	245.650200	3.405263	0.076842	-6.258418
Tables	58331.749216	648.794771	3.890282	0.261285	-55.565771

In [18]:

```
#Checking in the sub category field
dataset.groupby('Sub-Category').max()
```

Out[18]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sales	Quantity	Discount
Sub-Category											
Accessories	Standard Class	Home Office	United States	Yuma	Wisconsin	99207	West	Technology	3347.370	14	
Appliances	Standard Class	Home Office	United States	Wilmington	Wisconsin	99301	West	Office Supplies	2625.120	14	
Art	Standard Class	Home Office	United States	Yuma	Wisconsin	99301	West	Office Supplies	1113.024	14	
Binders	Standard Class	Home Office	United States	Yuma	Wisconsin	99301	West	Office Supplies	9892.740	14	
Bookcases	Standard Class	Home Office	United States	Wilmington	Wisconsin	98115	West	Furniture	4404.900	13	
Chairs	Standard Class	Home Office	United States	York	Wyoming	99301	West	Furniture	4416.174	14	
Copiers	Standard	Home	United States	Warwick	Washington	98198	West	Technology	17499.950	9	

Category	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sales	Quantity	Discount
Envelopes	Standard Class	Home Office	United States	Wilmington	Washington	98198	West	Office Supplies	604.656	9	
Category Fasteners	Standard Class	Home Office	United States	Westminster	Wisconsin	98632	West	Office Supplies	93.360	14	
Furnishings	Standard Class	Home Office	United States	Yonkers	Wisconsin	98661	West	Furniture	1336.440	14	
Labels	Standard Class	Home Office	United States	Woodland	Wisconsin	98115	West	Office Supplies	786.480	14	
Machines	Standard Class	Home Office	United States	Yuma	Washington	99207	West	Technology	22638.480	11	
Paper	Standard Class	Home Office	United States	York	Wisconsin	98502	West	Office Supplies	733.950	14	
Phones	Standard Class	Home Office	United States	Yonkers	Wisconsin	98661	West	Technology	4548.810	14	
Storage	Standard Class	Home Office	United States	York	Wisconsin	99301	West	Office Supplies	2934.330	14	
Supplies	Standard Class	Home Office	United States	Yonkers	Wisconsin	98115	West	Office Supplies	8187.650	10	
Tables	Standard Class	Home Office	United States	Yonkers	Wisconsin	99207	West	Furniture	4297.644	13	



In [19]:

```
#Checking in the sub category field
dataset.groupby('Sub-Category').min()
```

Out[19]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sales	Quantity	Discount	P
Sub-Category	First Class	Consumer	United States	Akron	Alabama	2038	Central	Technology	0.990	1	0.0	-75.0
Accessories	First Class	Consumer	United States	Akron	Alabama	2038	Central	Technology	0.990	1	0.0	-75.0
Appliances	First Class	Consumer	United States	Abilene	Alabama	1841	Central	Office Supplies	0.444	1	0.0	1181.0
Art	First Class	Consumer	United States	Akron	Alabama	1841	Central	Office Supplies	1.344	1	0.0	0.0
Binders	First Class	Consumer	United States	Akron	Alabama	1453	Central	Office Supplies	0.556	1	0.0	3701.0
Bookcases	First Class	Consumer	United States	Amarillo	Arizona	1040	Central	Furniture	35.490	1	0.0	1665.0
Chairs	First Class	Consumer	United States	Amarillo	Alabama	1810	Central	Furniture	26.640	1	0.0	-630.0
Copiers	First Class	Consumer	United States	Arlington	Alabama	1841	Central	Technology	299.990	1	0.0	59.0
Envelopes	First Class	Consumer	United States	Ann Arbor	Alabama	5408	Central	Office Supplies	1.632	1	0.0	0.0
Fasteners	First Class	Consumer	United States	Akron	Alabama	1841	Central	Office Supplies	1.240	1	0.0	-11.0
Furnishings	First Class	Consumer	United States	Akron	Alabama	1752	Central	Furniture	1.892	1	0.0	-427.0
Labels	First Class	Consumer	United States	Appleton	Alabama	1453	Central	Office Supplies	2.088	1	0.0	0.0
Machines	First Class	Consumer	United States	Arlington	Alabama	7060	Central	Technology	11.560	1	0.0	6599.0
	First Class	Consumer	United States					Office Supplies				

Paper	First Class	Consumer Segment	United States	Akron City	Alabama State	1453 Postal Code	Central Region	Office Supplies Category	3.380 Sales	1 Quantity	0.0 Discount	1.0 Profit
Phones Sub-Category	First Class	Consumer	United States	Akron	Alabama	1841	Central	Technology	2.970	1	0.0	-386.
Storage	First Class	Consumer	United States	Akron	Alabama	1453	Central	Office Supplies	4.464	1	0.0	-337.
Supplies	First Class	Consumer	United States	Aberdeen	Alabama	1841	Central	Office Supplies	1.744	1	0.0	1049.
Tables	First Class	Consumer	United States	Akron	Alabama	1841	Central	Furniture	24.368	1	0.0	1862.

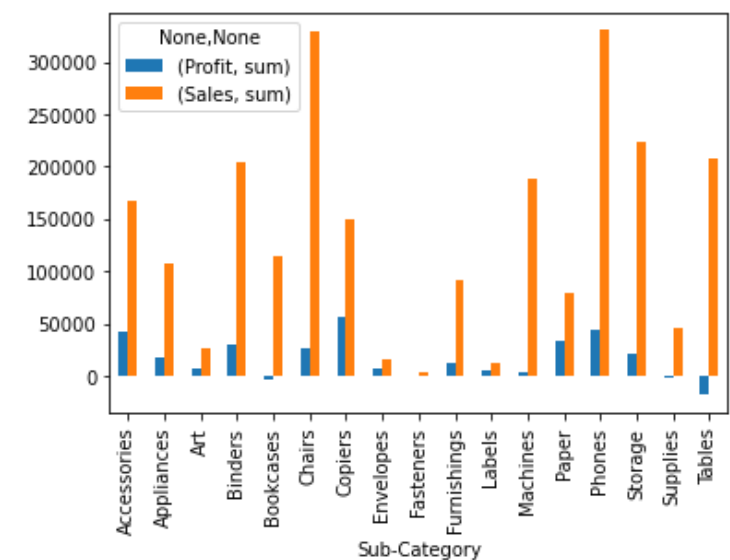
In [20]:

```
#for the sub category
plt.figure(figsize= (10,25))
dataset.groupby('Sub-Category')['Profit','Sales'].agg(['sum']).plot.bar()
plt.show()
```

<ipython-input-20-d72b26a4155c>:3: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
dataset.groupby('Sub-Category')['Profit','Sales'].agg(['sum']).plot.bar()
```

<Figure size 720x1800 with 0 Axes>



## Correlation

In [21]:

```
# Correlation in Dataset
corr=dataset.corr()
corr
```

Out[21]:

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961
Sales	-0.023854	1.000000	0.200795	-0.028190	0.479064
Quantity	0.012761	0.200795	1.000000	0.008623	0.066253
Discount	0.058443	-0.028190	0.008623	1.000000	-0.219487
Profit	-0.029961	0.479064	0.066253	-0.219487	1.000000

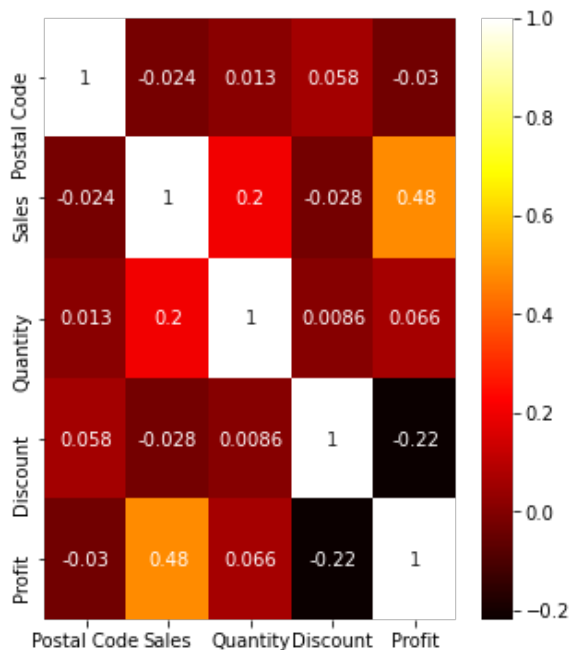
In [24]:

```
# Heat Map Visualization
```

```
plt.figure(figsize=(5,6))
sns.heatmap(corr,annot=True,cmap='hot')
```

Out[24]:

<AxesSubplot:>

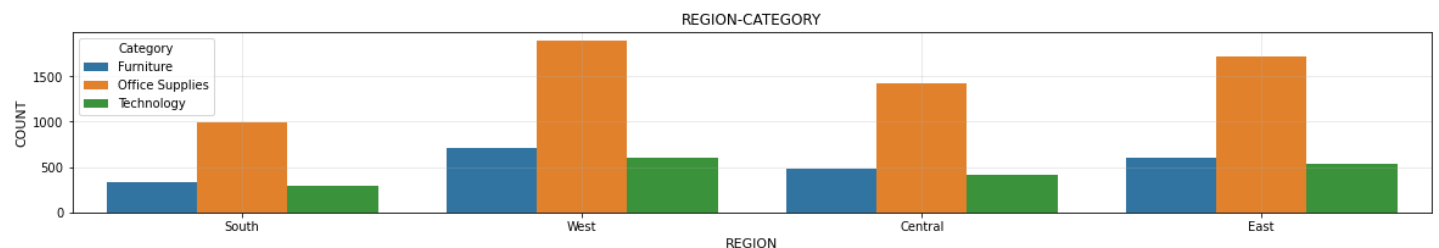


In [25]:

```
#Plotting based upon the REGION and the category based.
plt.figure(figsize=(20,6))
plt.subplot(2,1,1)
sns.countplot('Region',hue='Category',data=dataset)
plt.title('REGION-CATEGORY')
plt.ylabel('COUNT',fontsize=11)
plt.xlabel('REGION',fontsize=11)
plt.grid(alpha=0.3)
plt.show()
```

C:\Users\akell\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

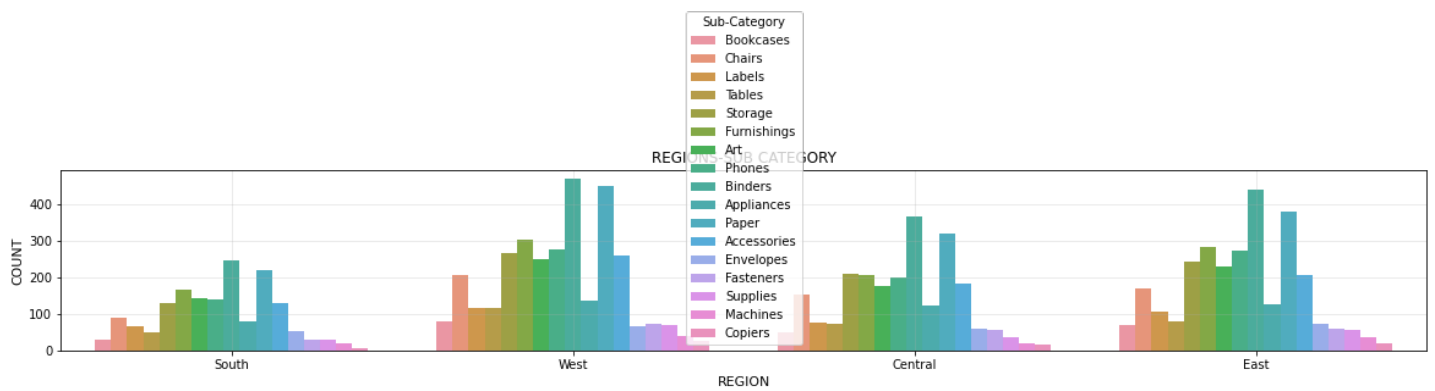


In [26]:

```
#Plotting based upon the REGION and the Sub-category based.
plt.figure(figsize=(20,6))
plt.subplot(2,1,2)
sns.countplot('Region',hue='Sub-Category',data=dataset)
plt.title('REGIONS-SUB CATEGORY')
plt.ylabel('COUNT',fontsize=11)
plt.xlabel('REGION',fontsize=11)
plt.grid(alpha=0.3)
plt.show()
```

C:\Users\akell\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn()
```

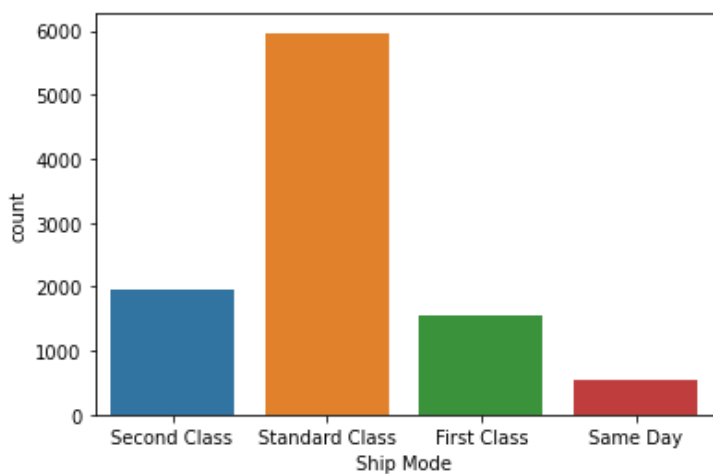


In [27]:

```
#Now the Analysis on the basis of shipping mode  
sns.countplot(x=dataset['Ship Mode'])
```

Out[27]:

<AxesSubplot:xlabel='Ship Mode', ylabel='count'>



In [28]:

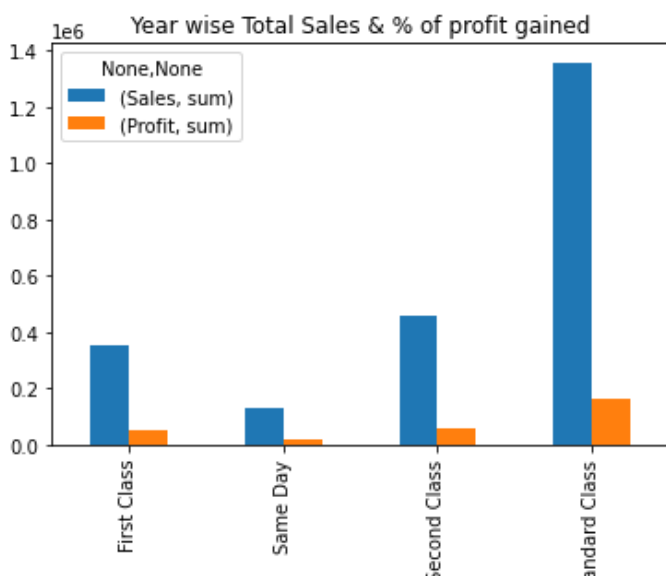
```
#Sales per Ship mode  
dataset.groupby('Ship Mode')['Sales', 'Profit'].agg(['sum']).plot.bar()  
plt.title('Year wise Total Sales & % of profit gained')
```

<ipython-input-28-e070edceff41>:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
dataset.groupby('Ship Mode')['Sales', 'Profit'].agg(['sum']).plot.bar()
```

Out[28]:

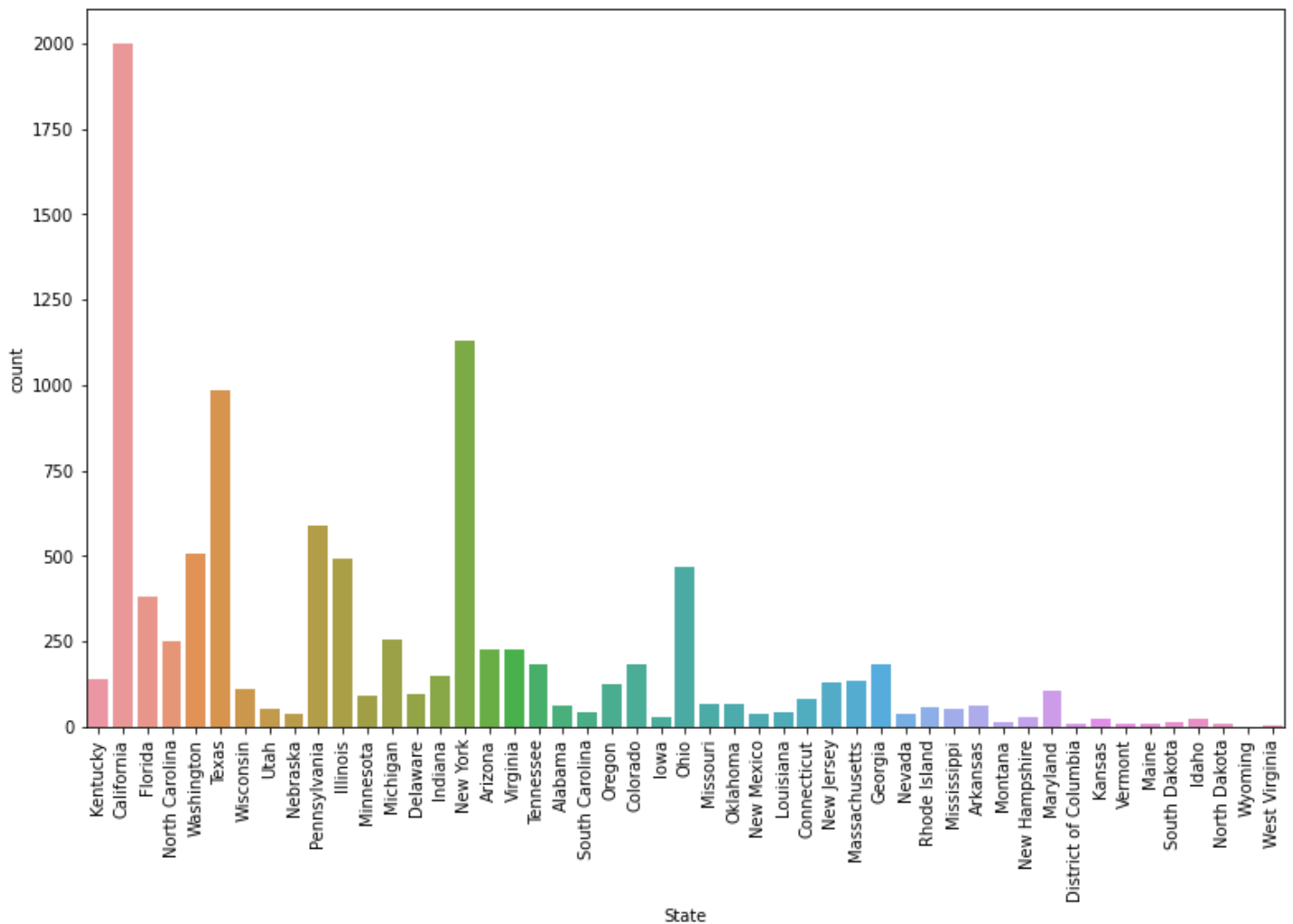
Text(0.5, 1.0, 'Year wise Total Sales & % of profit gained')





In [29]:

```
#Checking fo the state wise number of delivery.
plt.figure(figsize=(13,8))
sns.countplot(x=dataset['State'])
plt.xticks(rotation=90)
plt.show()
```



In [30]:

```
top_state=dataset.groupby('State').sum().sort_values('Profit',ascending=False)
top_state.head(15)
```

Out[30]:

	Postal Code	Sales	Quantity	Discount	Profit
State					
California	184382639	457687.6315	7667	145.6	76381.3871
New York	11835721	310876.2710	4224	62.4	74038.5486
Washington	49661687	138641.2700	1883	32.4	33402.6517
Michigan	12356103	76269.6140	946	1.8	24463.1876
Virginia	5124335	70636.7200	893	0.0	18597.9504
Indiana	6991602	53555.3600	578	0.0	18382.9363
Georgia	5685480	49095.8400	705	0.0	16250.0433
Kentucky	5725336	36591.7500	523	0.0	11199.6966
Minnesota	4932224	29863.1500	331	0.0	10823.1874
Delaware	1896504	27451.0690	367	0.6	9977.3748

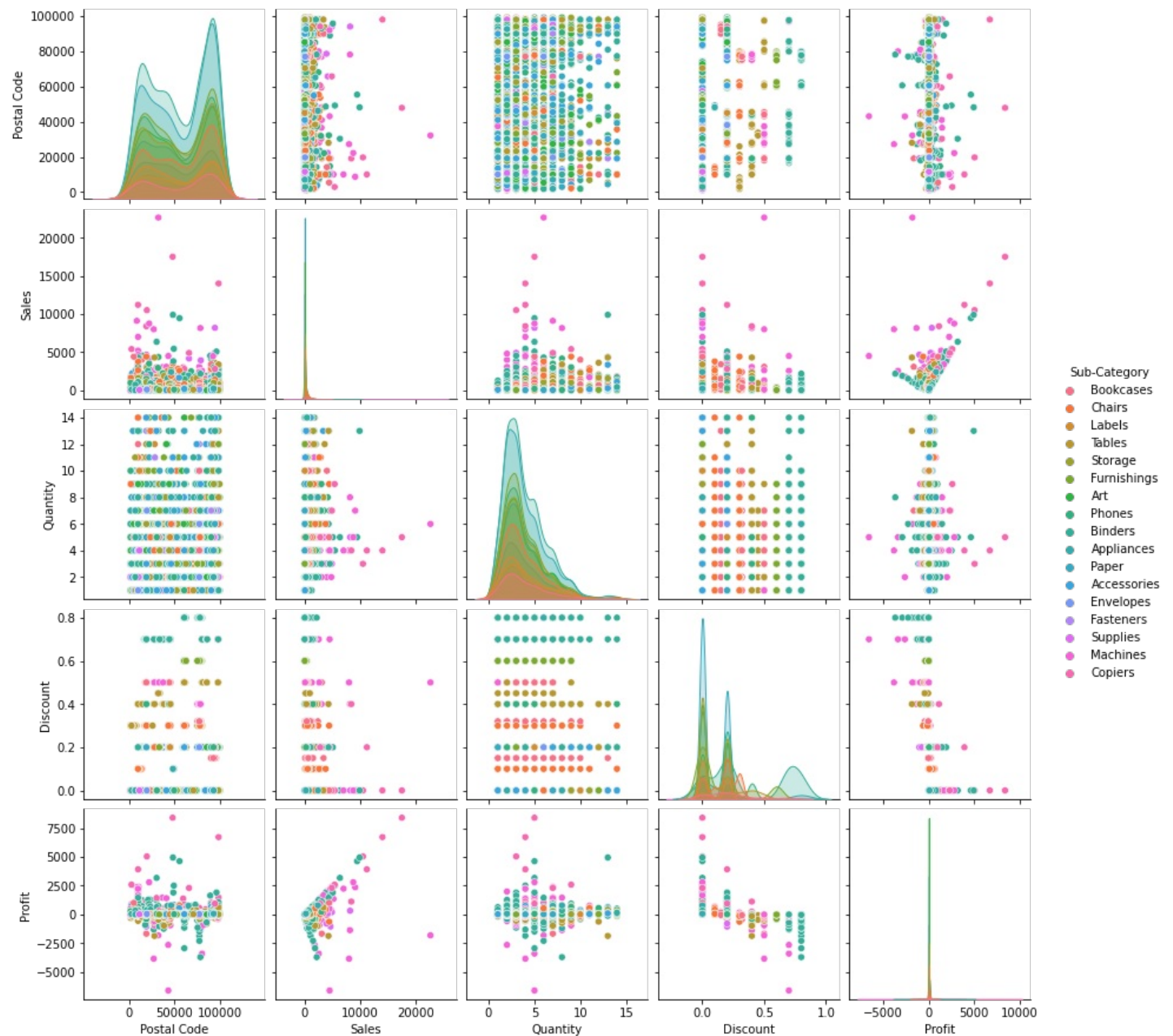
New Jersey	1007034	35764.8100	454	0.6	9772.9122
Wisconsin	5899704	32114.6100	463	0.0	8401.8004
Rhode Island	162878	22627.9560	199	1.2	7285.6293
Maryland	2206740	23705.5230	420	0.6	7031.1788
Massachusetts	268295	28634.4340	491	2.1	6785.5016

In [31]:

```
#Pairplot
figsize=(15,10)
sns.pairplot(dataset,hue='Sub-Category')
```

Out[31]:

<seaborn.axisgrid.PairGrid at 0x2b21e6a2b50>



In [ ]: